3. Compatibilities and Differences

The following subsections describe user and system administration issues involved in upgrading from a UNICOS 8.0 release to the UNICOS 9.0 release. This section also includes early information about changes planned for future UNICOS releases.

Because this release overview documents all features and compatibility issues introduced since the UNICOS 8.0 base release, each compatibility issue includes the UNICOS release level in which the compatibility issue was introduced. This information is provided to help our customers focus on the compatibility issues that will specifically affect their upgrade.

Each subsection in this section lists in the margin both the type of user and type of hardware affected. For definitions of the terms used, see subsection 1.6, page 1–6.

For information about UNICOS multilevel security (MLS) system compatibility issues, see subsection 4.3, page 4–9.

3.1 Compatibility statement

To meet user requirements for compatibility between releases of Cray Research software, upward compatibility is provided in subsequent releases of the system and products in the following areas:

- UNICOS user commands
- Standard language syntax, semantics, and Cray Research extensions
- Absolute binary code
- Relocatable binary code

This section describes any incompatible changes that were introduced in this release for the following reasons:

- Hardware changes
- Third-party software changes
- Improved software design or numerical techniques
- Bug fixes

If possible, both the old and new software are provided for one release. In other cases, compatibility bridging is provided through special compatibility software for the duration of one major release.

3.2 Selected hardware support withdrawn

Users affected

All

Supporting hardware

All Cray Research systems

The UNICOS 8.0 release was the final major release supporting the following Cray Research system hardware:

- CRAY X-MP EMA systems
- CRAY X-MP EA systems
- CRAY-2 systems
- I/O subsystem model B (IOS-B)
- IOS-C
- IOS-D
- Peripheral expander hardware

Support for use of the X-MP mode on CRAY Y-MP hardware has also been withdrawn.

> **Note:** When hardware support is dropped for a platform, support for compiling and linking for that platform is also dropped from all Cray Research platforms. For example, UNICOS 9.0 and subsequent releases do not support compiling, linking, or running CRAY X-MP binaries on CRAY Y-MP platforms. Additionally, commands such as debuggers and simulators will drop all support for binaries targeted for that platform.

For a list of hardware supported by the UNICOS 9.0 release, see subsection 7.1.1, page 7–1.

3.2.1 *Removal of IOS model D support*

Users affected

All

Supporting hardware

All Cray Research systems with IOS model Ds

Incompatibility introduced with release: UNICOS 9.0

With the UNICOS 9.0 release, IOS model D (IOS-D) systems are no longer supported. This support change will affect all sites with IOS-D systems. All commands and command options specific to the IOS-D have been removed from the UNICOS 9.0 release.

The following commands have been removed for the UNICOS 9.0 release:

- `bconfig`(8)
- `bgdiag`(8)
- `cleario`(8)
- `disk`(8)
- `donut`(8)
- `dsdiag`(8)
- `eft`(8)
- `flaw`(8)
- `idmp`(8)
- `mfdump`(8)
- `netconf`(8)
- `olcfdt`(8)
- `surf`(8)
- `tp`(4)

The following commands have been changed for the UNICOS 9.0 release:

- `brc`(8): `DUMPADEV` and `DUMPBDEV` variables removed
- `cpdmp`(8): `-f` option removed
- `dsk`(4): IOS-D functionality removed
- `hippi`(4): software loopback feature removed, IOS-D `loctl` calls not valid
- `olnet`(8): `FSL`, `FST`, and `HSX` tests removed

3.3 StorageTek support changes

Users affected

All

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 9.0

Prior to UNICOS 9.0, users of the EMASS robotic and ER90 device products had to special-order them. With UNICOS 9.0, both products are included in the UNICOS release materials.

For additional information on licensing these products, see subsection 7.2.3.6, page 7–10. For additional information on ER90 device support for the CRAY EL series, see subsection 2.3.6, page 2–12.

Beginning with the UNICOS 9.0 release, if you use an automated cartridge library from Storage Technology Corporation (StorageTek) attached to a Sun system in conjunction with your Cray Research system, you must do the following:

- Upgrade to ACSLS version 5.0 (provided by Storage Technology Corporation)
- If you have Redwood drives, order Cray Research license CRSTK/STKRED

You will also find `stknet` files in a new location.

3.3.1 *ACSLS 5.0 required*

Users affected

All

Supporting hardware

All Cray Research systems

ACSLS 5.0 from StorageTek uses packet structure version 4. The `stknet` executable program that will run with UNICOS 9.0 on the Cray Research system will use packet structure version 4 as defined in ACSLS 5.0; Therefore, you must upgrade to ACSLS 5.0. The version of `stknet` that will run with UNICOS 9.0 will work only with ACSLS 5.0 and not with older packet structures defined in earlier versions of ACSLS.

3.3.2 *Order CRSTK/STKRED*

Users affected

All

Supporting hardware

All Cray Research systems

If ACSLS serves a StorageTek library equipped with Redwood drives and you use `stknet` to mount tape cartridges on those Redwood drives, you must have a Cray Research CRSTK/STKRED license. With this license, you will receive a Cray FLEXlm key, which enables stknet to operate with the Redwood drives.

If you use a StorageTek library that does not use Redwood drives, no Cray license is required, but you must use ACSLS 5.0 for the UNICOS 9.0 support to function properly.

For licensing information for StorageTek Redwood Drives with UNICOS 9.0, see subsection 7.2.3.4, page 7–9, or contact your Cray Research representative.

3.3.3 *New* `cmd/c1/tp/stkacs` *subdirectory*

Users affected

All

Supporting hardware

All Cray Research systems

With UNICOS 9.0, source code for `stknet.c` and the associated header files will no longer be available and will be removed from the `cmd/c1/tp` directory.

The components for `stkacs` will reside in a new directory; the components are the `Nmakefile` module, relocatable module `stknet.o`, and archive `stklib.a`. These components will be distributed with the UNICOS 9.0 release materials to all sites.

3.4 Networking and communications

The following subsections describe compatibility issues affecting network connectivity and communications.

3.4.1 *Compatibility concerns for NQS, NQE, and FTA*

The following subsections describe compatibility issues affecting Network Queuing System (NQS), Network Queuing Environment (NQE), and File Transfer Agent (FTA).

3.4.1.1 *NQS changes*

Users affected

All

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 8.0.3/8.3

The availability of a Network Queuing Environment (NQE) for UNICOS has caused changes to the Network Queuing System (NQS) configuration.

> **Note:** The following changes affect NQS itself, whether or not you are using NQE.

NQE consists of two existing UNICOS products (NQS and the File Transfer Agent (FTA)) and one additional, separately licensed product, Network Queuing EXtensions (NQX).

NQS now requires the file `/etc/nqeinfo`. This file contains configuration information that NQS and NQX use on startup. This file is placed in `/etc` by the installation process. The UNICOS Installation/Configuration Menu System tool uses an input template `nqeinfo` file that is located in `/usr/src/skl/etc`. You can use the menu system to ensure that the values in `nqeinfo` are appropriate for your installation. NQS will not start without an appropriately configured `/etc/nqeinfo` file.

The `/etc/nqeinfo` file generation is performed automatically, and by default, by the menu system tool. Sites that want to install NQS without NQX must leave `NQE configuration` enabled in the configuration menu.

The following menu selections affect NQE configuration. The selections are shown with their default settings.

The option to configure NQE appears under the menu for `Configurator Automation Options`, as follows:

```
S->  NQE configuration?                    YES
```

The option to enable NQE appears under the menu for `Major Software Configuration`, as follows:

```
S->  Network Queueing Environment (NQE)   ON
```

To start NQS without NQX, use the `qstart`(8) command. To start both NQS and NQX, use the `nqeinit`(8) command. If you disable `CONFIG_NQE`, the `nqeinfo` file is not built during installation, and NQS will not start. If for some reason you do not have an `nqeinfo` file after installing, you can copy the default `nqeinfo` file to `/etc` from `/usr/src/skl/etc/nqeinfo`.

Although NQX requires a separate license, the other components of NQE for UNICOS (NQS and FTA) do not require licenses. They remain part of the UNICOS product set. For information about NQX licensing, see subsection 7.2.3.1, page 7–8.

3.4.1.2 `qdump/qload` *utility scripts not required for NQS 9.0*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 9.0

Beginning with the UNICOS 9.0 release, the `qdump/qload` utility scripts are no longer needed. The documentation for `qdump` and `qload` has been removed from the UNICOS 9.0 manuals. These scripts were originally provided to upgrade from UNICOS 7.0 to UNICOS 8.0, due to incompatibilities in the NQS spool directory between these two releases. These commands are no longer needed for upgrading from UNICOS 8.0 releases to UNICOS 9.0 because the NQS spool directory provided with UNICOS 8.0 is upward compatible with the NQS spool directory provided with UNICOS 9.0.

Jobs submitted at the UNICOS 8.0 release level will be rerun after the upgrade to UNICOS 9.0.

Restart files are incompatible between UNICOS 8.0 and UNICOS 9.0. Batch jobs that are checkpointed on a UNICOS 8.0 system are not automatically restarted on a UNICOS 9.0 system.

Although the NQS spool directory is upward compatible from UNICOS 8.0 to UNICOS 9.0, backward compatibility is neither guaranteed nor supported. While testing the UNICOS 9.0 release, it is recommended that you use a different `/usr/spool/nqs` directory be used, to preserve the

UNICOS 8.0 NQS database. Note that the spool directory is specified by the `NQE_NQS_SPOOL` variable in the `/etc/nqeinfo` file, so it is not necessary to rebuild NQS to change the NQS spool directory. See subsection 3.4.1.3.1 for more information.

If after starting production with UNICOS 9.0 it becomes necessary to run UNICOS 8.0, use one of the following procedures:

- Let the queues empty as jobs finish and then fall back to 8.0.
- Perform the following steps:

  Stop all queues so that no more queued NQS requests are initiated.

  Disable the queues so that no more requests are submitted.

  Rerun all running jobs (see the `qmgr rerun request` command). Jobs that can be rerun will be queued. Jobs that can not be rerun will continue running; You can either terminate these jobs or let them finish running.

  Move queued jobs to an enabled, stopped pipe queue on another NQS or NQE host (see the `qmgr move` command). Users must have accounts on the host to which the queued jobs are routed.

  Wait for running jobs and jobs submitted by the UNICOS station call processor (USCP) to finish, or terminate them.

  Reload the UNICOS 8.0 system (using a different NQS spool directory), and then route the queued jobs.

If you have any questions about either method, contact your Cray Research service representative.

3.4.1.3 *NQS and FTA compatibility concerns*

The following subsections describe NQS and FTA compatibility concerns.

3.4.1.3.1 `/etc/nqeinfo` added for NQS and FTA configuration

Users affected

Administrator

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 8.0.3/8.3

NQS and FTA now use the `/etc/nqeinfo` file for configuration information at start-up time. NQS and FTA will not start without an appropriately configured `/etc/nqeinfo` file. To ensure that the values in `nqeinfo` are appropriate for your installation, use the NQE menu in the UNICOS Installation/Configuration Menu System.

Configuration parameters in the `/etc/nqeinfo` file will override equivalent NQS configuration parameters in the `/usr/src/net/nqs/include/config.h` file. To see the values NQS is currently using, use the `qconfigchk`(8) command.

For more information about the `/etc/nqeinfo` file, see the *UNICOS NQS and NQE Administrator's Guide*, SG–2305.

3.4.1.3.2 NQS `qstart` and `qstop` commands location

Users affected

Administrator

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 9.0

In the UNICOS 9.0 release, the `qstart`(8) and `qstop`(8) commands are installed in the `/etc` directory instead of the `/usr/bin` directory.

3.4.1.3.3 NQS `qstart` and `qstop` commands source

Users affected

Administrator

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 8.0.3

Beginning with the UNICOS 8.0.3 release, the `qstart`(8) command is installed from `/usr/src/net/nqe/bin/qstart`. Previously, it was installed from `/usr/src/net/nqs/cmd/qstart.sh`. The `qstop`(8) command is installed from `/usr/src/net/nqe/bin/qstop`. Previously, it was installed from `/usr/src/net/nqs/cmd/qstop.sh`.

If you have local mods to `qstart`(8) or `qstop`(8), the mods will no longer apply because `qstart` and `qstop` have been moved from the NQS program library (PL) to the NQE source tree. Because an NQE UNICOS source manager (USM) PL does not exist, you can change `qstart` and `qstop` by either editing their scripts or creating your own USM PL.

3.4.1.4 *NQE job dependency support for UNICOS 9.0 systems*

Users affected

All

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 9.0

If you use the Network Queung Environment (NQE) job dependency feature, you cannot run redundant NQE master servers. If a master server goes down, it can not synchronize the job dependency information stored in the Network Load Balancer (NLB) server database with events that may have occurred when the machine was down.

3.4.1.5 *Compatibilty between NQX and NQE 1.x and NQX 2.0 NLB master server*

Users affected

All

Supporting hardware

All Cray Research systems

NQX/NQE 1.*x* `cload`(1) clients and certain NQX/NQE 1.*x* `nlbconfig` command options require updating to function with an NQX 2.0 Network Load Balancer (NLB) master server.

If you have NQE/NQX 1.*x* `cload`(1) clients that need to communicate with an NQX 2.0 NLB server, then you must make changes to the NQX 2.0 NLB server `name_map` file. In the definition of the NJS object, comment out all of the attributes inclusively between:

```
NJS_ORIG_SEQNO  "Request sequence number"
integer(20);

NJS_RPRIORITY  "URM priority incr (qsub -p)"
integer(207);
```

When all `cload`(1) clients are upgraded to the NQE 2.0 level, the `name_map` file should be changed back.

Neither the NJS name list nor the NJS description list can exceed 4096 bytes. It is recommended that you remove all of the status API additions (the status API is available only in NQE 2.0, and will be available for NQX 3.0) for completeness. Note also that there are approximately 190 attributes that will be removed.

3.4.1.6 `cqstat` *display now prompts for password*

Users affected

Administrator, End user

Supporting hardware

All Cray Research systems

The `cqstat`(1) display now will prompt for a password when a user requests detailed job status on a server where password validation is turned on.

However, the password is displayed on the user's screen as the user enters it. This problem will be fixed in NQX 3.0. An alternative for requesting full status, in this case, is to use the –P option to the `cqstatl`(1) command.

3.4.1.7 *Final release supporting the NQS* `config.h` *file*

Users affected

Administrator

Supporting hardware

All Cray Research systems

The UNICOS 9.2 restricted release will be the last UNICOS release to support the NQS `config.h` file. Beginning with the UNICOS 10.0 major release, the NQS `/usr/src/net/nqe/src/nqs/include/config.h` file will not be available. The configuration settings currently in the `config.h` file will either be moved to the `/etc/nqeinfo` file or will no longer be configurable. The `/etc/nqeinfo` file provides the capability to customize NQS without recompiling NQS programs. The settings within the `/etc/nqeinfo` are available at run time.

This change is being made to support Cray Research's continuing move towards binary releases, to provide a more consistent interface for UNICOS NQS and CraySoft NQE, and to simplify NQS customization.

3.4.2 *New format for the* `gated.conf` *file*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 8.3

The `gated.conf` file has a new format that is incompatible with previous releases.

Configure the `gated.conf` file through the UNICOS Installation/Configuration Menu System by using the following menu selection:

```
Configure System

   Network configuration
      TCP/IP configuration
         Routing
```

For more information, see the `gated-config`(5) man page.

Related publications

UNICOS Networking Facilities Administrator's Guide, publication SG–2304

3.4.3 `route` *command* `-C` *option removed*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 8.3

As announced in an earlier UNICOS release, the `-C` option has been removed from the `route`(8) command. The `-C` option forced the `route`(8) command to use the old `ioctl` interface to the kernel for routing operations.

3.4.4 *BGP, EGP, and* `HELLO` *protocols no longer supported with* `gated`

Users affected

Administrator

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 8.3

The `gated`(8) daemon no longer supports the Border Gateway Protocol (BGP), Exterior Gateway Protocol (EGP), and `HELLO` protocol.

3.4.5 *Integrated DCE login*

Users affected

All

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 8.0.4

If your system has the Distributed Computing Environment (DCE version 1.0.2.1 or later) installed and turned on, you no longer need to log in to DCE in addition to logging in to the UNICOS system. Integrated DCE login lets users log in to a Cray Research system and have DCE identities (credentials) set up transparently (that is, they will not have to use the `dce_login` command to perform the DCE login process).

The integrated DCE login feature is a supplemental authentication mechanism that lets users access the Cray Research system and simultaneously have a DCE identity created. This feature supports the `rlogin`(1b), `telnet`(1b), `rexec`(3), `ftp`(1b), `fta`(8), and `nqe`(7) commands. To

authenticate users in the DCE registry, the integrated DCE login feature requires a password (currently, Cray DCE/Distributed File System (DFS) does not support ticket forwarding or inheriting). Because of the password restriction, the integrated DCE login does not support `.rhosts` or `host.equiv` functionality for `rlogin`(1b) and `rcmd`(8). For `rlogin`(1b) requests in which the remote user has a `.rhost` file set up, DCE authentication will not occur. DCE authentication is passive (it is not grounds for rejecting a user login). So the user will be allowed on the system but will not have a DCE identity set up. To use DCE/DFS services as an authenticated user, the user then must perform a separate `dce_login`.

For security reasons, a user who logs in with the `root` user name will not be authenticated; to gain access to DCE services, they must perform a `dce_login`.

3.4.6 *DECnet application no longer supported on CRAY EL systems*

Users affected

End user, administrator, operator

Supporting hardware

CRAY EL systems

Incompatibility introduced with release: UNICOS 9.0

The UNICOS 8.0 release was the final major release supporting the DECnet application for CRAY EL systems; beginning with the UNICOS 9.0 release, it is no longer supported. The DECnet application, kiNET, from Ki Research is a third-party application that used the character special file `/dev/ge`, which allowed you to run the DECnet protocols over Ethernet on a CRAY EL system.

3.4.7 *HYPERchannel no longer supported on CRAY EL systems*

Users affected

All

Supporting hardware

CRAY EL series

Incompatibility introduced with release: UNICOS 9.0

The UNICOS 8.0 release was the final major release supporting HYPERchannel on CRAY EL systems; beginning with the UNICOS 9.0 release it is no longer supported.

3.4.8 *Final release supporting USCP and Stations*

Users affected

All

Supporting hardware

All Cray Research systems except CRAY T90, CRAY J90, and CRAY EL series

Incompatibility introduced with release: UNICOS 10.0

UNICOS 9.0 is the final major release supporting Cray Research Station products and the UNICOS station call processor (USCP). USCP and the following station products will, however, be supported until one year after UNICOS 10.0 is released: IBM MVS Station, IBM VM Station, DEC VAX/VMS Station, and CDC NOS/VE Station. TCP/IP-based alternatives to stations are available for all of these environments through a combination of Cray Research and third-party vendor products.

For more information on station migration options, contact your Cray Research representative.

3.4.9 *Final release supporting OSI*

Users affected

Users of the OSI optional product

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 10.0

UNICOS 9.0 is the final major release supporting the Open Systems Interconnection (OSI) protocol family, AF_ISO, at the socket level. Support for OSI lower layers in UNICOS and for the OSI unbundled product will not be available beginning with the UNICOS 10.0 release.

3.4.10 *Final release supporting `netmon`, `xnetmon`, and `xsnmpmon` networking tools*

Users affected

All

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 10.0

UNICOS 9.0 is the final major release supporting the following networking tools: `netmon`, `xnetmon` and `xsnmpmon`. Beginning with the UNICOS 10.0 release, these tools will not be supported.

To provide standard monitoring of new and existing devices, there will be new simple network management protocol (SNMP) software that will support standard management information bases (MIBs) for these industry devices. However, Cray Research will not be providing any user displays providing

monitoring capabilities. These features will, therefore, be available only from a third-party network management station provided by the customer.

For further information contact your Cray Research representative.

3.4.11 *UNICOS 9.2 is final release supporting full X11 clients*

Users affected

All

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 10.0

Cray Research ports X11 from the X Consortium, Inc. distribution of the X11 product. The Consortium has decided to discontinue support of a number of programs as of X11 release 6. Therefore, Cray Research will also no longer support the affected features with X11R6, which will be released with UNICOS 10.0.

The following clients will still be available to the public in the X Consortium `contrib` directory:

Client	Description
`listres`	List resources in widgets
`viewres`	Graphical class browser for Xt
`xcalc`	Scientific calculator for Xt
`xditview`	Display distroff output
`xdpr`	Dump an X window directly to a printer
`xedit`	Simple text editor for X window system
`xfontsel`	Point and click interface for selecting X11 fonts
`xload`	System load average display for X window system
`xpr`	Print an X window dump

The `contrib` directory will remain available through anonymous `ftp` at `ftp.x.org`.

Note: The X Consortium is also discontinuing support for xman, a manual page display for the X Window System. Cray Research will continue to support xman in the UNICOS 10.0 major release and in the Cray Visualization Toolkit (CVT) release.

3.4.12 *New* `dnsquery` *command for domain name servers*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 8.3

The new `dnsquery`(8) command provides an interface to name servers by using Berkeley Internet name domain (BIND) resolver library calls. `dnsquery`(8) supports queries to the name server with an operation code of `QUERY`. This command is intended to be a replacement or supplement to commands such as `nstest`, `nsquery`, and `nslookup`(1).

For more information on `dnsquery`(8), including syntax and options, see the `dnsquery`(8) man page.

3.5 User interface, tools, commands, and utilities

The following subsections describe compatibility issues affecting the user interface, tools, commands, and utilities.

3.5.1 *UNICOS tools and libraries moved to the programming environment*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Change introduced with release: UNICOS 8.3

The Fortran libraries and most UNICOS performance utilities are no longer being released with the UNICOS system. Instead, they are released with the programming environment releases, beginning with the 1.2 releases.

> **Note:** Sites installing the UNICOS 9.0 release must upgrade to the programming environment 1.2 or later releases to obtain Fortran run-time libraries.

The `libf`, `libu`, and `libfi` Fortran libraries and related commands and include files were moved from the UNICOS operating system to the CrayLibs portion of the programming environments. All routines in these libraries are now released and documented as part of the programming environments. The following commands are also released and documented with the programming environments:

- `asgcmd`(1)
- `assign`(1)

- `fdcp`(1)

These utilities are released with CrayLibs, which is part of the programming environment releases.

The following performance utilities were moved from the UNICOS operating system to the CrayTools portion of the programming environments. All of these utilities are now released and documented as part of the programming environments.

- `atchop`(1)
- `atexpert`(1)
- `flowview`(1)
- `jt`(1)
- `jumpview`(1)
- `libperf`
- `libprof`
- `libtrace`
- `perfdmp`(1)
- `perfview`(1)
- `procview`(1)
- `prof`(1)
- `profview`(1)

These utilities are documented in the *Tuning Guide to Parallel Vector Applications*, publication SG–2182. The *UNICOS Performance Utilities Reference Manual*, publication SR–2040, is now obsolete.

3.5.2 *Standard shell ability to execute* `setuid` *and* `setgid` *scripts removed*

Users affected

All

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 8.3

In order to support the UNICOS multilevel security (MLS) policy, the standard shell no longer will execute `setuid` and `setgid` shell scripts. Validation of `setuid` and `setgid` programs now is performed exclusively by the kernel. This change is consistent with the majority of computer vendors.

3.5.3 *New search options added to* `man` *utility, obsolete section title option*

Users affected

All

Supporting hardware

All Cray Research systems

Initial release: UNICOS 9.0

The `-s` and `-l` options were added to the `man`(1) utility. The new search options make the `man` *section title* option obsolete. The `man` *section title* option will be removed in a future release.

3.5.4 *New* `scanit` *command for CRAY J90 scalar cache use*

Users affected

All

Supporting hardware

CRAY J90 systems

Initial release: UNICOS 8.0.4A

Scalar cache was disabled in the UNICOS 8.0.3.2J restricted release because of a problem found when a certain sequence of instructions was executed with scalar cache enabled. The problem caused incorrect calculations, data corruption, and (ORE) aborts.

The new `scanit`(1) command will search executables for code sequences that cause errors when scalar cache is enabled. The `scanit`(1) command changes the code so the sequence of instructions does not cause errors. The `scanit`(1) command may be used on other vendor supplied programs only if the vendor permits program modification.

For more information, see the `scanit`(1) man page, or contact your Cray Research representative.

3.5.5 *Checkpoint/restart incompatibility*

Users affected

All

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 8.3

Because of features introduced in this UNICOS release, checkpoint files created on a UNICOS 8.0 system cannot be restarted under UNICOS 9.0.

3.5.6 New bc *utility replaces previous* bc *and* dc *utilities*

Users affected

End user

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 8.3

The bc(1) and the dc(1) utilities included in previous UNICOS releases were renamed to be obc(1) and odc(1), respectively, in the UNICOS 9.0 release. The obc(1) and odc(1) commands will be removed in a future UNICOS release.

The new bc(1) utility uses source code ported from the Free Software Foundation (FSF). This new bc(1) utility is different from the version of bc(1) provided with previous UNICOS releases in the following ways:

- The dc(1) utility is not invoked as part of the processing done by bc(1); the new bc(1) utility incorporates the functionality of the bc(1) and the dc(1) utilities that were included in previous UNICOS releases.
- The `-c` option, which produced compile only output, has been removed.
- The new `-s` option causes bc(1) to accept only the language defined by POSIX.
- The new `-w` option warns of the use of extensions to the POSIX bc utility.
- Names can be longer than 1 character.
- POSIX `if` statements do not allow an `else` clause.
- Logical operators, a `read` function, a `print` statement, and a `continue` statement have been added.
- Numbers cannot contain embedded spaces.
- The source code for bc(1) is now included with the release.

The bc(1) man page from previous releases is now available under the name obc(1). The man page for the FSF utility is available under the name bc(1).

3.5.7 *The* make *command sends* .f *and* .F *files directly to* cf77

Users affected

End user

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 8.3

The make utility has improved the Fortran source preprocessing by invoking a Fortran preprocessor. In prior UNICOS releases, the make(1) command had rules for converting .F files to .f files by calling the CF77PP preprocessor (/lib/cpp). The default rules of the make(1) command have been modified to call the cf77(1) command, which automatically calls the gpp preprocessor (/usr/lib/gpp) for .F input files. The CF77PP and CF77PPFLAGS macros are no longer defined by the make(1) utility. To pass options to the cf77 preprocessor, the FFLAGS macro can be used.

The following differences exist between the gpp and cpp preprocessors:

- The gpp preprocessor does not process #define macros unless the -F option is set. Macros are not processed by default because they can cause errors by expanding Fortran source lines to more than 80 characters.
- The gpp preprocessor does not process C-style comments If allowed, gpp could confuse Fortran code or Fortran comments as C-style comments, causing errors.
- The gpp preprocessor does not process #pragma, #line, #class, or #ident directives because these directives are used only in C.
- The gpp preprocessor does not process the -C, -E, or -N options.
- The cpp preprocessor does not process the -F or -Q options.
- The cpp preprocessor cannot correctly interpret some Fortran code.
- The cpp preprocessor cannot add the BTRNSFRM and ETRNSFRM directives.

If you want to continue to have make(1) call cpp for .F files, you must call /lib/cpp explicitly or write your own make(1) rule.

3.5.8 `sar` *utility changes*

Users affected

End user, administrator

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 8.3

The following changes have been made to the system activity reporting command, `sar`(1):

- The display of the `-X` option has been altered to show the total abnormal exchanges, rather than a rate that is almost always 0.
- The `ldcache` statistics are reset to 0 upon reconfiguration of the system `ldcache` parameters.

3.5.9 *Support removed for several UNICOS utilities*

Users affected

All

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 8.3

Several UNICOS utilities have been removed from the UNICOS system, as announced in the *UNICOS 8.0 Release Overview*, publication RO–5000 8.0. The utilities removed are as follows:

- `cl`(1)
- `ftref`(1)
- `hconv`(1)
- `ldovl`(1)
- `lmedit`(1)
- `lmstak`(1)
- `nicem`(8)
- `nmodex`(1)
- `nupdate`(1)
- `oawk`(1)
- `old`(1)
- `osh`(1)
- `pcpp`(1)
- `plcopy`(1)
- `pshell`(1)
- `whodo`(8)

3.5.10 *Support removed for* nupdate *utility*

Users affected

Programmer

Supporting hardware

All Cray Research systems

Incompatiblilty introduced with release: UNICOS 9.0.

The `nupdate` utility has been removed in UNICOS 9.0 so that support and development efforts can be focused on only one source control utility.

To prepare for the removal of the `nupdate` utility, convert existing `nupdate` program libraries (PLs) to USM PLs before you upgrade to UNICOS 9.0. USM can convert existing `nupdate` PLs to USM PLs; however, because the `nupdate` utility may be used in the conversion process, you may not be able to convert them on any UNICOS releases after UNICOS 8.0. For information on how to convert `nupdate` PLs to USM PLs, see the *UNICOS Source Manager (USM) User's Guide*, publication SG–2097.

The Source Code Control System (SCCS), which is available in UNICOS 9.0, is an alternative for source code maintenance. Unlike USM, SCCS is portable across UNIX systems. UNICOS SCCS utilities conform to the X/Open Portability Guide, version 4 (XPG4) standard and are compatible with all other X/Open compliant systems.

3.5.11 *GNU Emacs upgraded to Emacs-19.28*

Users affected

End user

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 8.3

As announced in the *UNICOS 8.0 Release Overview*, publication RO–5000 8.0, GNU Emacs-19 has replaced Emacs-18.

The following subsections describe differences that exist between the GNU Emacs-19 and GNU Emacs-18 user start-up file, `.emacs`. If the differences affect you, you must modify your `.emacs` file. You will, however, be able to construct the start-up file so that only parts of the file will execute depending on which version of Emacs is running. Refer to the sample start-up file `/usr/lib/emacs/etc/.emacs`.

3.5.11.1 *Defining keys for editing functions*

The way that users define keys for editing functions in the `.emacs` file has changed.

For GNU Emacs-18, users must use actual keyboard sequences to identify the keys. For example:

```
define-key RSI-map "226z" 'query-replace) ;;  F3
```

For GNU Emacs-19, users can name keys according to what appears on the key tops. For example:

```
(define-key function-key-map [f14] [undo]) ;; L4
```

If a user's `.emacs` file defines key functions according to the old GNU Emacs-18 method, the user must change the `.emacs` file according to the new GNU Emacs-19 method. Starting GNU Emacs-19 with a `.emacs` file that contains old definitions will be successful, but the key function definitions will not take effect.

3.5.11.2 *New GNU utilities*

GNU Emacs-19 includes GNU utilities that were not part of the GNU Emacs-18 release. For GNU Emacs-18, users can obtain these utilities from the Free Software Foundation, make them work on the UNICOS system, and use the `.emacs` file to load the utilities into Emacs. For GNU Emacs-19, using the `.emacs` file to load utilities that are already part of release 19 will cause errors.

3.5.12 *UNICOS 9.0 and SecurID version 2.3 dependency*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 9.0

Customers who use SecurID and enable the SecurID password checking option must upgrade to version 2.3 or later of SecurID. When the password checking option is enabled, the user is authenticated against both the SecurID passcode and the user database (UDB) password. When the password checking option is disabled, authentication is performed only against the SecurID passcode.

UNICOS 9.0 does not support the SecurID password checking option on versions of SecurID prior to version 2.3. When using SecurID, only passcode authentication will be performed, regardless of the password checking option configuration.

3.5.13 *UNICOS 9.0 and SMARTE 5.0 dependency*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 9.0

Customers who use system maintenance and remote testing environment (SMARTE) must upgrade to version 5.0 or later with the UNICOS 9.0 release. UNICOS 9.0 does not support earlier versions of SMARTE software.

3.5.14 *Final release including the Docview utilities*

Users affected

All

Supporting hardware

All Cray Research systems

The UNICOS 9.0 release is the final major release for which documentation will be delivered in Docview format. You will be able to access documents already in Docview format until software support for the UNICOS 9.0 release ends.

Cray Research now provides online versions of manuals through the CrayDoc online information reader, a workstation-based electronic viewing tool that includes graphics, hypertext linking, and both book and multibook searches.

3.5.15 *Final release supporting the process monitor,* `xproc`

Users affected

End user

Supporting hardware

All Cray Research systems

The UNICOS 9.0 release is the final major release supporting the process monitor, `xproc`(1). The `xproc`(1) utility provides a visual tool, based on the X Window System, that displays and controls UNICOS processes and NQS jobs. Users who install NQE will be able to display and control NQS jobs. The `ps`(1) and `kill`(1) commands can be used to display and control UNICOS processes.

3.5.16 *Final release supporting the file manager,* `xfm`

Users affected

End user

Supporting hardware

All Cray Research systems

The UNICOS 9.0 release is the final major release supporting the File Manager, `xfm`(1). The `xfm`(1) utility provides a visual tool, based on the X Window System, that performs file management for the UNICOS system. When `xfm`(1) is retired, there will be no X Window System-based file manager tool on UNICOS. In the future, the Common Desktop Environment (CDE) will provide a file manager as part of the desktop; however, CDE may not be available at the time that `xfm`(1) is retired.

3.5.17 `diskmap` *man page renamed to be* `diskspec`

Users affected

All

Supporting hardware

All Cray Research systems

Change introduced with release: UNICOS 9.0

The `diskmap`(7) man page was renamed to be `diskspec`(7) to more accurately reflect its content, which is disk specifications.

3.5.18 *UNICOS 9.2 to package CAL with programming environment*

Users affected

Administrators

Supporting hardware

All Cray Research systems

Packaging change introduced with release: UNICOS 9.2

UNICOS 9.1 will be the last UNICOS release to include Cray Assembly Language (CAL). Beginning with the release of UNICOS 9.2, CAL will be available on the programming environment release media. This is a packaging change only; CAL will still be licensed with UNICOS and there will be no additional fees charged beyond the cost for the UNICOS license.

3.6 System calls

The following subsections describe compatibility issues that affect system calls.

3.6.1 `sigctl` *system call functionality restricted*

Users affected

Programmer

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 8.3

With the introduction of the `sigaction`(2) system call in the UNICOS 6.0 release, the `sigctl`(2) system call has become obsolete. While `sigaction`(2) does not provide a superset of the functionality of `sigctl`(2), the additional functionality that `sigctl`(2) provides is no longer considered necessary.

The specific additional functionality provided by `sigctl`(2) is the ability to choose an arbitrary action (such as termination with or without a core dump) for an arbitrary signal. In contrast, `sigaction`(2) only allows the user to specify a signal handler, ignore a signal, or accept a system-defined, signal-specific default action.

This additional functionality in `sigctl`(2) was provided so pre-existing interfaces, such as `signal`(2), could be implemented in terms of `sigctl`(2) without `sigctl`(2) setting default action policy. The main value of the `sigctl`(2) system call to Cray Research users was to provide a signal registration function that did not revert to a default action after entering a signal handler; this behavior is now provided by `sigaction`(2).

The functionality of the `sigctl`(2) system call is now restricted. Any attempt to specify an action other than to catch (`SCTL_REG`) or ignore (`SCTL_IGN`) is silently mapped to the system-defined default action for the signal being registered. To avoid confusion, a new action flag for `sigctl`(2) has been provided. Making the `sigctl(SCTL_DEF,` *signo*`, 0)` call causes the default action to be taken when the signal *signo* is received.

3.6.2 `acctctl` *accounting system call replaces* `acct` *and* `dacct` *system calls*

Users affected

Administrator, programmer

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 8.3

The new `acctctl`(2) system call enables, disables, and checks the status for process, daemon, and record accounting. The `acctctl`(2) system call replaces the `acct`(2) and `dacct`(2) system calls.

3.6.3 `limits` *system call functionality changes with addition of* `policy` *system call*

Users affected

Administrator, programmer

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 8.3

The functionality of the `limits`(2) system call is now limited to lnode operations. The `policy`(2) system call handles other fair-share functionality. The new `policy`(2) system call provides an interface to the kernel for the fair-share daemon, and allows access to the fair-share constants structure, `sh_consts`.

The `L_GETCOSTS` and `L_SETCOSTS` functions of the `limits`(2) system call have been removed in the UNICOS 9.0 release. The `policy`(2) system call provides the `get_costs` and `set_costs` actions as replacements.

3.7 Libraries

The following subsections describe compatibility issues that affect run-time library routines.

3.7.1 *UDB library changes*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 8.3

The `libudb`(3) library replaces the `udblib` library. The library routines for the user database (UDB) have been moved from `libc/gen` to `libc/udb`.

3.7.2 *Changes to memory manager routines*

Users affected

Programmer

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 8.3

The memory manager routines formerly documented on the `memmgr`(3) man page have been replaced with a new set of routines documented on the `malloc`(3) man page. The old routine names may be removed in a future release of the UNICOS system. The old routines and their corresponding replacements are as follows:

Old routine	New routine
`_memcheck`	`malloc_check`
`_inplace`	`malloc_inplace`
`_expand`	`malloc_expand`
`_extend`	`malloc_extend`
`_howbig`	`malloc_howbig`
`_isvalid`	`malloc_isvalid`
`_memfree`	`malloc_space`
`_memlimit`	`malloc_limit`
`_memstats`, `memstats`	`malloc_stats`
`_memtron`	`malloc_tron`
`_memtroff`	`malloc_troff`
`_memetrac`	`malloc_etrace`
`_memdtrac`	`malloc_dtrace`
`_malloc`	none; use `malloc`
`_free`	none; use `free`
`_realloc`	none; use `realloc`

The new routines are similar to the old routines, except that their names are in the user's name space (they do not begin with an underscore), and all sizes are in bytes rather than in words.

For complete information on the new routines, see the `malloc`(3) man page. The `memmgr`(3) man page has been removed.

3.8 Operating system

The following subsections describe compatibility issues that affect the operating system.

3.8.1 *UNICOS non-MLS and MLS systems to be merged as of UNICOS 10.0 release*

Users affected

End user, administrator

Supporting hardware

All Cray Research systems

On UNICOS 8.0 and UNICOS 9.0 systems, the MLS configuration is optional. That is, to use security features specific to the UNICOS MLS configuration, sites must enable the `SECURE_CONFIG` configuration parameter.

As of the UNICOS 10.0 release, the `SECURE_CONFIG` parameter will not be supported, and MLS features will be incorporated into the UNICOS 10.0 system.

Combining the non-MLS and MLS systems makes the security features, such as access control lists (ACLs) and security auditing, available to all Cray Research customers. Note that use of these security features will still be optional. Sites can choose to use only those features that are applicable to their sites.

For sites that are currently using a UNICOS non-MLS system, this feature will provide a UNICOS configuration that has the same behavior as the current UNICOS non-MLS system. For sites that are currently using a UNICOS MLS system, this feature will provide a UNICOS configuration that has the same behavior as the current UNICOS MLS system. Commands, libraries, and system calls will not change. However, interfaces that are currently available only on UNICOS MLS systems will be generally available on UNICOS 10.0 systems.

Changes will be made to some of the default settings of options in the UNICOS Installation/Configuration Menu System. These changes will not impact sites that import their previous configurations.

For more information about other UNICOS 10.0 security feature changes, see subsection 4.3.4, page 4–10.

3.8.2 `gencat` *utility renamed to be* `mfsck`

Users affected

Administrator

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 8.3

The parallel file system check utility formerly known as `gencat`(8) has been renamed to be `mfsck`(8). Scripts and procedures must be changed to use the `mfsck`(8) name.

This change was required to comply with the XPG4 standard, which uses the name `gencat` for a message system utility.

For more information, see subsection 3.9.3, page 3–38.

3.8.3 *Fair-share scheduler differences*

Users affected

Administrator

Supporting hardware

All Cray Research systems

This UNICOS release introduces the following fair-share scheduler differences:

- Incompatibility introduced with release: UNICOS 8.0.3/8.3

  The `-y` option of the `shradmin`(8) command is no longer deferred; the fair-share scheduler now allows administrators to set charges for terminal I/O operations.

  For more information, see the `shradmin`(8) man page.

- Incompatibility introduced with release: UNICOS 8.3

  A feature was added to enforce correct usage of the fair-share flags (the `shrflags` field) in the user database (UDB). If incorrect information in the UDB would cause the creation of an improperly labeled or positioned lnode, the system issues an error and the user is prevented from logging in or submitting jobs. The major impact of this feature will be felt by sites that run with Share by Account mode enabled. The `SHAREHOLDER` flag must be set in the `shflags` field for UDB records that define account IDs (acids).

  For more information, see the `shrtree`(8) and `shradmin`(8) man pages.

- Incompatibility introduced with release: UNICOS 8.3

  As announced in the *UNICOS 8.0 Release Overview*, publication RO–5000 8.0, the commands `shrates`(1), `shrinfo`(1), `shrusage`(1), and `shrstats`(1) have been deleted. Their functionality has been replaced by the `shrview`(1) command, which was introduced in UNICOS 8.0. The `shrmon`(8) command, however, remains for compatibility reasons; it will be deleted in a future release. Users of `shrmon` are encouraged to convert to `shrview`, which will continue to be supported in future releases.

  For more information, see the `shrview`(1) and `shrmon`(8) man pages.

3.8.4 *Programming considerations for dynamic allocation of file descriptors*

Users affected

Programmer

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 8.3

The following programming recommendations are related to the dynamic allocation of file descriptors feature in the UNICOS 9.0 release. For a description of the feature, see subsection 2.9.19, page 2–73.

- The dynamic allocation of file descriptors feature in the UNICOS 9.0 release will not affect existing binaries. However, use caution when writing new code that makes use of this feature.

  Some existing binaries could use unsafe programming practices or make assumptions about `OPEN_MAX` having a value of 64 and being the maximum number of files a UNICOS process can open. An example of this would be direct use of the `OPEN_MAX` define to allocate file-related storage or to loop through file descriptors. If a process opens more than `OPEN_MAX` files and then uses `exec`(2) to execute such a binary, the execution of that binary may fail.

  The `_sc_open_max` parameter to `sysconf`(2) should be used instead to avoid this problem. This recommendation also applies to UNICOS 8.0, where `OPEN_MAX` has a value of 255.

- When a process dynamically increases its open files, the open file array of the process is no longer assured to be contained in the internal `ucomm` structure of the process. Programs making use of the `proc`(4) `PRFS_UCOMM` function should use the new `PRFS_FDS` function to get the open file array of such a process. Use of `PRFS_FDS` is needed only when looking at processes that have opened more than 300 (`K_OPEN_DEFAULT`) files.

3.8.5 *Default number of current primary partitions changed*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 8.3

The limit on the number of primary partitions in a file system that are maintained with current data has been set at the default value of 4. The `fsck`(8) command has been modified to account for this change.

When an administrator or operator is switching to a file system created with an older version of the UNICOS system, the following messages may be encountered:

```
/scr_401: file system opened
/scr_401: super block fname scr_401, fpack scr_400
/scr_401: alternate dynamic block in block 300027
differs from primary
/scr_401: rewrite dynamic blocks ('y' or 'n')?
```

These messages may occur multiple times for one file system, and for multiple file systems. If the administrator answers `y` (yes), the file system is restored to its older state under which all primary partitions are active.

If the administrator answers `n` (no), the update occurs during most normal file system activity.

Caution: Answering `n` could cause a serious problem in the unusual case that all active primary partitions are down, and the file system was not checked. In this case, the old dynamic data and bit map could lead to damage of existing data.

3.8.6 *UDB files*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 8.3

The user database (UDB) has been substantially rewritten to allow for future enhancements. The UDB has been extended and now spans several files. There are now extension files in a new subdirectory, `/etc/udb_2`, that physically separate public and private data in the UDB. (Private data includes encrypted passwords, security compartments, permission bits, and so on.) Fields added appear in the extension files, with the exception of the user name, UID, group list, and account ID (acid) list.

> **Note:** The files `/etc/udb` and `/etc/udb.public` continue to exist for compatibility; however, in a future release they will be removed and all data will be moved into the new files.

For UDB library compatibility issues, see subsection 3.7.1, page 3–27. For Fair-share compatibility issues, see subsection 3.8.3, page 3–30.

3.8.7 *New `relasync` command added*

Users affected

Administrator, system analyst

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 9.0

The binary packaging tools are being upgraded, resulting in the removal of the `ldproto`(8) command. The `ldproto`(8) command will be replaced by a more robust, more easily maintainable set of tools that can be used by both UNICOS and asynchronous products.

The UNICOS release introduces the first of these new tools with the `relasync`(8) command. The `relasync`(8) command verifies loaded files and directories on disk against information in a database.

For more information, see the `relasync`(8) man page.

3.8.8 `dodisk` *mail output removed*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 8.3

The `dodisk`(8) accounting command no longer sends mail to the administrator that shows the current date and time. This change brings `dodisk`(8) into conformance with other UNIX implementations.

For more information, see the `dodisk`(8) man page.

3.8.9 *UNICOS SFS* `esdaemon` *command replaced by* `sfsd` *command*

Users affected

Administrator

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 9.0

The `sfsd`(8) command, which initializes and monitors the external semaphore device for the UNICOS shared file system (SFS) feature, has replaced the `esdaemon`(8) command.

For more information, see the `esdaemon`(8) man page.

3.8.10 *Character-special tape interface node 43 (ddal) support removed*

Users affected

End user, administrator, operator

Supporting hardware

CRAY J90 series and CRAY EL series

Incompatibility introduced with release: UNICOS 9.0

The character-special tape interface that uses major node number 43 (which is also referred to as the `ddal` interface), is no longer supported. As of the UNICOS 9.0 release, the character-special tape interface created by the `tpdaemon`(8) command is the only character-special tape interface supported.

3.8.11 *UNICOS tape subsystem becomes a separate product at UNICOS 10.0*

Users affected

All

Supporting hardware

All Cray Research systems

Packaging change introduced with release: UNICOS 10.0

Beginning with the UNICOS 10.0 release, the UNICOS tape subsystem will be packaged with the release materials in binary form and will be sold as a separate product for new hardware platforms. Users of CRAY T3E systems and subsequent new platforms must purchase a license and obtain a FLEXlm key to enable the subsystem to function.

The new product will contain the feature-rich tape daemon-assisted interface; the character-special tape interface will continue to be available without cost as part of the UNICOS release materials. Its online tape capabilities will provide tape access similar to the access that users on other UNIX systems have. This access, a basic means of reading and writing tape information, will not require a license or a FLEXlm key.

3.8.12 *UNICOS 8.0 release support ending*

Users affected

All

Supporting hardware

All Cray Research systems

Support for the UNICOS 8.0 release will end on September 30, 1996.

Note: For model D systems only, UNICOS 8.0 and its related products will be supported with updates as needed from September 30, 1996, through September 30, 1997.

3.8.13 `mtdump` *and* `segldr` *packaging change*

Users affected

All

Supporting hardware

All Cray Research systems

The UNICOS 9.0 major release will be the final release containing both source and binary code for `mtdump` and `segldr`. Beginning with the UNICOS 9.1 release, `mtdump` and `segldr` will be packaged in binary form only. The generation compiler necessary to build these utilities will not be part of the UNICOS release package.

3.9 XPG4 compliance

Changes have been made to make the UNICOS system compliant with the X/Open XPG4 standard. The following subsections describe the UNICOS 9.0 release XPG4 compliance and compatibility issues.

3.9.1 *Function prototypes changed for compliance with XPG4*

Users affected

Programmer

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 8.3

Several function prototypes in various system include files have been changed to conform with the XPG4 standard. Some of these new prototypes may require users to recompile or change source code that built successfully in previous releases.

The include files and prototypes in the following list have changed. The include file name, the old prototype, and the new prototype are specified for each changed prototype.

File: <nl_types.h>
Old: `char *catmsgfmt(const char *, const char *, const int, const char *, const char *, char *, const int)`
New: `char *catmsgfmt(const char *, const char *, int, const char *, const char *, char *, int)`

File: <nl_types.h>
Old: `char *catgetmsg(const nl_catd, const int, const int, char *, const int)`
New: `char *catgetmsg(nl_catd, int, int, char *, int)`

File: <nl_types.h>
Old: `char *catgets(const nl_catd, const int, const int, const char *)`
New: `char *catgets(nl_catd, int, int, const char *)`

File: <nl_types.h>
Old: `nl_catd catopen(const char *, const int)`
New: `nl_catd catopen(const char *, int)`

File: <nl_types.h>
Old: `int catclose(const nl_catd)`
New: `int catclose(nl_catd)`

File: <search.h>:
Old: void *lsearch(const void *, const void *, size_t, size_t, int (*)(const void *, const void *))
New: void *lsearch(const void *, void *, size_t, size_t, int (*)(const void *, const void *))

File: <search.h>:
Old: void twalk(void *, void (*)(void *, VISIT, int)
New: void twalk(const void *, void (*)(const void *, VISIT, int)

File: <stdlib.h>:
Old: char *getpass(char *, ...)
New: char *getpass(const char *)

File: <stdlib.h>:
Old: int putenv(char *)
New: int putenv(const char *)

File: <stdio.h>:
Old: long getw(FILE *)
New: int getw(FILE *)

File: <stdio.h>:
Old: long putw(long, FILE *)
New: int putw(int, FILE *)

File: <string.h>:
Old: void swab(const char *, char *, int)
New: void swab(const void *, void *, ssize_t)

File: <time.h>:
Old: time_t timezone;
New: long timezone;

File: <unistd.h>:
Old: int chroot(char *)
New: int chroot(const char *)

File: <unistd.h>:
Old: char *getpass(char *, ...)
New: char *getpass(const char *)

File: <unistd.h>:
Old: void swab(const char *, char *, int)
New: void swab(const void *, void *, ssize_t)

File: `<unistd.h>`:
Old: `int ulimit(int, long)`
New: `int ulimit(int, ...)`

3.9.2 *XPG4 utilities now use compliant message system*

Users affected

End user

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 8.3

UNICOS user-level utilities subject to the XPG4 standard now issue messages using the XPG4-conformant UNICOS message system. In prior releases, each utility used internal messages. As a result of this, the format of messages has been changed. This also provided an opportunity to clarify messages and make them more helpful.

These changes support the `NLSPATH` environment variable specified in the XPG4 standard. The language and cultural conventions of diagnostic and information messages is affected by `LC_MESSAGES` and `NLSPATH` settings.

The format of utility messages is controlled by a combination of the new `CMDMSG_FMT` environment variable and the existing `MSG_FORMAT` environment variable.

For more information on messages, see the `explain`(1) man page.

Related publications

- *UNICOS Message System Programmer's Guide*, publication SG–2121

3.9.3 `gencat` *and* `catgen` *utilities renamed for compliance with XPG4*

Users affected

End user, administrator

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 8.3

To comply with the XPG4 standard, the following utilities were renamed:

- The message system administrator utility formerly named `catgen`(8) has been renamed to be `gencat`(1). (This utility is used to rebuild UNICOS message system catalogs.) As a compatibility bridge, the utility will be callable through the

name `catgen` in the 9.0 release and 9.0 revision releases. Using the `catgen` name will produce an informational message. As of the UNICOS 10.0 release, only the name `gencat` will be recognized.

- Because the `catgen`(8) command has been renamed to be `gencat`(1), the file system utility formerly known as `gencat`(8) has been renamed to be `mfsck`(8). (This utility runs parallel file system checks.) Scripts and procedures must be changed to use the `mfsck`(8) name.

The operation of these utilities is not affected by the name changes.

For more information, see the `gencat`(1) and `mfsck`(8) man pages.

3.9.4 *Commands changed for compliance with XPG4*

Users affected

End user, programmer

Supporting hardware

All Cray Research systems

The following commands were changed to conform to the XPG4 standard. Subsections 2.10.5, 2.10.6, 2.10.7, and 2.10.8 list all user commands, system calls, library routines, and external variables that comply with the XPG4 standard.

For more information about these utilities, see their man pages.

Note: One of the central purposes for the UNICOS 9.0 major release is to deliver an XPG4-compliant environment. This required many changes to commands. With each change made, Cray Research made every effort to identify and minimize the impact on users of prior releases (release-to-release compatibility). This section describes the changes necessary for XPG4 compliance that are likely to affect users.

While testing for compliance, the new XPG4 commands and utilities test suite (with more than 6000 tests) also identified a number of latent defects not related to the XPG4 standard. Although we have not identified any such cases, procedures that depend on the defective behavior may be affected by the corrective code.

- Incompatibility introduced with release: UNICOS 8.3

  For the nm(1) utility, the -g option (print only global symbols) replaces the -f option (which was redundant with -g in previous releases). The -A option replaces the -o option as the method for causing the file and member name to be prepended to the symbol name.

- Incompatibility introduced with release: UNICOS 8.3

  For the od(1) utility, the default output has been changed to specify 2-byte words. The od(1) command formerly specified its output in 8-byte words. Use the -to option to receive output in 8-byte words.

- Incompatibility introduced with release: UNICOS 8.3

  For the pr(1) utility, the -L option performs the function (folding lines of output) formerly performed by the -f option. The -f option is now used to specify that a form-feed character should be used for new pages instead of the default sequence of newline characters.

- Incompatibility introduced with release: UNICOS 9.0

  The pr(1) utility options -l, -w, and -o now require the argument with the option; if the argument is not present, an error message will be returned.

- Incompatibility introduced with release: UNICOS 8.3

  The output of the ps utility when the -f or -l options are specified has a new column labeled C. This column does not have meaning on a UNICOS system, but its presence is required by the standard.

- Incompatibility introduced with release: UNICOS 9.0

  The cd(1) utility now writes the name of the new working directory only to standard output when it is called from within a script. Users who use the cd(1) utility through the CDPATH environment variable, or with a dash (–) as the directory operand from within a shell script, will notice this change. If you do not want the name of the new working directory to appear in the output, redirect the output from the cd(1) utility as follows: `cd - > /dev/null`. This change only applies to cd(1) when called from within a script; there is no change to cd(1) in interactive mode.

- Incompatibility introduced with release: UNICOS 9.0

  The `sccs`(1) utility options `admin`, `delta`, and `prs` now require the argument with the option; if the argument is not present, an error message will be returned.

- Incompatibility introduced with release: UNICOS 9.0

  The shell built-in `time` command was renamed to be `shtime` so that when the `time` command is entered, the behavior and output format match the behavior and output format of the `/bin/time` command and comply with the X/Open XPG4 standard. Anyone using the shell built-in `time` command must use `shtime`; anyone using the `/bin/time` command will see no change.

- Incompatibility introduced with release: UNICOS 9.0

  The shell (`sh` and `ksh`) changed as follows for XPG4 compliance:

  1. When the `trap` built-in command is invoked with no options or operands, output is now `trap -- 'echo foo' INT`, instead of `trap 'echo foo' INT`. This change may affect anyone who depends on the output of the `trap` command.

  2. When displaying the status of stopped jobs, the `jobs` built-in command now also indicates the name of the signal that caused it to stop. This change may affect anyone who depends on the output of the `jobs` command and tries to parse the output positionally, which is not commonly done from shell scripts.

     For example:

     ```
     [1] + Stopped (SIGTSTP)        sleep 30
     ```

  3. Parameter expansion using expressions such as `${var:-`*word*`}`, where `var` is an unset or null variable, result in *word* being expanded, and now requires using the tilde expansion in addition to parameter expansion, command substitution, and arithmetic expansion. For example, in the case of `${var:-~tee/foo}`, when `var` is unset or null, the XPG4 compliant shell expands it to `/cool/u6/tee/foo`. This change could affect the behavior of shell scripts if they expect that tildes will not be expanded. In most cases, the results will be the same,

since it is likely that the tilde will be expanded later anyway. If you really do not want a tilde to be expanded, you must escape it. For example:

```
% echo ${var:-~tee/foo}
/cool/u6/tee/foo
% echo ${var:-\~tee/foo}
~tee/foo
```

4. The positional parameter `0` should be unchanged by a call to a function. On a call to a function, `$0` now remains unchanged instead of being reset to the name of the function. This change could affect anyone who checks `$0` from inside a function and expects that it has been changed to the function name.

5. A variable assignment specified with a special built-in command remains in effect after the built-in completes.

   For example:

   ```
   var=value set -a
   ```

   Because `set` is a special built-in command, the variable assignment `var=`*value* now remains in effect in the current shell invocation after the `set -a` command completes. This is a very uncommon thing to do with special built-in commands (`break`, `continue`, `return`, `eval`, `set`, `trap`, and so on), but if you do it, you could be changing your current environment while thinking that you are only affecting the environment of the command.

6. A word expansion or redirection error during a call to a function now aborts a noninteractive shell. If you have a word expansion or redirection error in a function in your script, then you will have problems. The desired behavior is probably to abort as soon as the error is detected.

7. The `AND` and `OR` operators now have equal precedence and are evaluated by the shell from beginning to end (left to right). This change will be noticed by anyone who uses commands lists using the `AND` (`&&`) and `OR` (`||`) operators and does not group them in any way.

8. When the value of the internal field separator (IFS) consists of a combination of white-space characters and other characters, then any sequence of zero or more of the IFS white-space characters with a single occurrence of one

of the other IFS characters now serves to delimit a field. This is a very minor change that could affect any shell script that defines its own IFS characters and expects multiple contiguous IFS characters (which must be a mixture of white-space and nonwhite-space characters) to delimit null fields.

9. When in command mode and the command ?*string*<newline> is entered, where *string* is found in the current command, the cursor now moves to the beginning of that string. Previously, searching with ? did not look for the string on the current line. This change will be noticed by anyone who uses command-line editing in an interactive shell. It will not affect shell scripts.

10. When a job is suspended, it becomes the current job. When at least two jobs are suspended, they become both the current and previous jobs. This will affect anyone who has stopped jobs, uses the job control symbols %+ and %–, and expects that the last job that was put in the background is guaranteed to be %+. It will not affect the use of %1, %2, and so on.

- Incompatibility introduced with release: UNICOS 9.0

  An interpretation request was submitted regarding the following change to verify it is in conformance with the standard; if it is not, the change will not be made.

  The order in which input records are parsed into fields has been changed in the awk(1) command. Previously, input records were not parsed into fields until a field was required. With this change, input records are parsed into fields as soon as they are read, which can change the way that the fields are parsed when the action statement of the awk program assigns the FS (field separator) variable. The FS variable is the field separator. For example, prior to this change, if *awkfile* contained:

  ```
  a.b.c d.e.f
  ```

  then the following script would print a:

  ```
  awk '{FS = "."} {print $1}' awkfile
  ```

  With this change, the same script prints a.b.c.

Users may change their scripts to obtain the old behavior of awk. A suggested change is as follows:

```
awk -F"." '{print $1}' awkfile
```

or:

```
awk 'BEGIN {FS = "."} {print $1}' awkfile
```

Both of these scripts will work with both the new and old versions. This change could impact anyone whose awk scripts set the FS variable.

- Incompatibility introduced with release: UNICOS 9.0

  The vi(1) command was changed as follows for XPG4 compliance.

  Note: The following changes were made to comply with POSIX 1003.2 (and XPG4). Due to complaints about the POSIX 1003.2 definition of vi, POSIX is currently rewriting its vi definition to take some previous behavior into account. It is possible that in the future some of the following changes may need to be reversed to comply with POSIX 1003.2b. Because of this, these changes were implemented in a way that gives users the option of turning off the POSIX behavior (all except the [*count*] e change, which corrects a bug) and using the previous behavior. To use the previous behavior, use the vi option (no)posix92. You can turn it ON/OFF from within vi by entering the following:

  ```
  set posix92          /* ON */
  set noposix92        /* OFF */
  ```

  It will be ON by default.

  You can also add set noposix92 to your $HOME/.exrc file or your EXINIT environment variable if you always want the previous behavior.

  – If both the -t *tagstring* and -c *command* options are given, the -t *tagstring* option is processed first. This should have very little effect on anyone; it adds more functionality.

  – When scroll down is entered in command mode and the line movement would place the current line after the end of the file, then the movement does not occur, the terminal is

alerted, and the current position is not changed. Anyone who uses `vi` very much will be sure to notice this change.

- When `u` is entered in command mode and the reversal only affects one line, then `vi` positions the cursor at the first nonblank character of the changed line. Anyone who uses `vi` very much will be sure to notice this change.
- When deleted text is placed in the unnamed buffer by using the `S` command, it also is placed in the numbered buffers. This change adds new functionality.
- Cursor positioning was incorrect when using the `p` command to put a buffer that contains multiple incomplete lines. This is a very minor change because this is a very uncommon action.
- The [*count*] `e` was incorrect when the cursor is already at the end of a word. This was a bug in `vi` that was fixed.
- The [*count*] `.` command now repeats the last command that changed the buffer. Previously, this did not include the `&`, `u`, and `U` commands, and some cases of the `p` and `P` commands. Anyone who uses `vi` very much will be sure to notice this change.
- The `vi` command was not handling paragraphs correctly. The characters `{` and `\f` at the beginning of line can now also be used to delimit paragraphs. This will affect anyone who expects paragraphs to be delimited only by blank lines.

- Incompatibility introduced with release: UNICOS 9.0

  The `lex`(1) command was changed as follows for XPG4 compliance:

  new `lex`(1): `^foo |bar = ^(foo|bar)`
  old `lex`(1): `^foo|bar = (^foo)|bar`

  Users may need to change `.l` files that included the `lex`(1) command.

- Incompatibility introduced with release: UNICOS 9.0

  The XPG4 standard required three changes to the `yacc`(1) parser-generator utility. The effects of these changes are minor for users; very few `yacc` (*.y) files required modification, and those that did required only trivial changes.

XPG4 requires the programs section's contents be included in the code file after the tables and code generated by `yacc`. Previously, the programs section was inserted before code generated by `yacc`.

Note: The above change may cause compile-time warnings or errors. These compilation errors are easy to fix. Typically, function prototypes will be required.

XPG4 requires certain global variables be of type "int." Some of the required variables were of type "long." This will affect only programs using these global variables in code that was not generated by `yacc`.

The retyped variables are `yydebug`, `yychar`, `yyerrflg`, `yynerrs`, `yytabelem`, `yystate`, `yys`, `yyps`, and `yytmp`. Programmers using these global variables will have to change their code to match code generated by `yacc`.

For example, change:

```
long yydebug;
```

to:

```
int yydebug;
```

Lastly, XPG4 makes it illegal to assign the same value to two different tokens. In the past, duplicate tokens were accepted and ignored, potentially masking problems in user code.

3.9.5 `locale` *definition support removed for compliance with XPG4*

Users affected

End user, programmer

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 9.0

Due to XPG4 compliance in the UNICOS 9.0 release, customers can no longer define their own locales in the manner described on the `locale`(1) man page. The new `localedef`(1) command provides a superset of this functionality.

Previously constructed locales will continue to be supported by the `setlocale`() interface until the UNICOS 10.0 release.

3.10 POSIX Threads (Pthreads) support

Users affected

Programmer

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 8.3

The UNICOS process model is now consistent with the process model described by the POSIX Threads (Pthreads) interface. The key areas involved in the changes are the definition of a process with respect to multitasking and the behavior of signals in a multitasked application.

These changes do not affect single-tasked applications; the new semantics are detectable only from multitasked applications. Many of these changes take affect only if applications are recompiled. However, some behaviors also affect existing binaries. The following subsections describe the specific incompatibilities and their scope.

3.10.1 *Multitasked process model changes*

Users affected

Programmer

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 8.3

As was first presented in the UNICOS 8.0 release documentation, the definition of a *process* in the UNICOS system is changing for multitasked applications. Previously, the UNICOS system supported a model in which a *multitasking group* described a shared address space in which multiple processes executed. In the new model, a process (or multitasking group) contains multiple *light-weight processes* (LWPs).

The primary change to support this new process model is in those system calls that support process IDs as part of their interface. In particular, the following new system calls were provided in UNICOS 8.0:

- `_getlwpid`(2)
- `_getlwppid`(2)
- `_lwp_exit`(2)
- `newgetpid`(2)
- `newgetppid`(2)
- `_newexit`(2)

The first three system calls were provided as synonyms for `getpid`(2), `getppid`(2), and `_exit`(2). It was also stated at that time that `getpid`(2), `getppid`(2), and `_exit`(2) would be changed and that the last three system calls provided early access to the new behavior.

In UNICOS 9.0, the behavior of `getpid`(2), `getppid`(2), and `_exit`(2) has been changed to that of `newgetpid`(2), `newgetppid`(2), and `_newexit`(2), respectively. (Note that the behavior of `exit`(3) has changed similarly to `_exit`(2).)

All of these changes affect only applications that are recompiled under UNICOS 9.0.

3.10.2 *Multitasked signal changes*

Users affected

End user, programmer

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 8.3

The changes to signals are largely the result of the process model changes described previously. The basic idea is that a multitasking group is the only entity that can be signaled from outside of that multitasking group.

In order to support this new signal model, the `kill`(2) and `killm`(2) system calls do not support the sending of signals to a specific light-weight process (LWP) in a multitasking group. Instead, the system chooses one of the LWPs to service the signal.

This change of behavior does not have any adverse effects on macrotasked or Autotasking applications because such applications have no specific control over the behavior of LWPs. However, the change affects applications that use the `tfork`(3) and `t_fork`(3) functions because tasks and LWPs have a one-to-one mapping in those applications.

This change affects only signals that are registered to execute a signal handler. If the signal has a system-defined default action associated with it, the behavior (whether to terminate or create a core dump) continues to affect all LWPs. Additionally, the `tfork`(3) and `t_fork`(3) functions support individual registration for each thread. Since the `kill`(2) and `killm`(2) system calls take action based on the registration of a signal, they must choose an arbitrary thread to reflect the action taken by the entire multitasking group. Because of this, `kill`(2) and `killm`(2) should not be used to send signals to such applications.

To allow a signal to be used with these applications, the `_lwp_kill`(2) and `_lwp_killm`(2) systems calls provide the `kill`(2) and `killm`(2) behavior that occurred in previous UNICOS releases; however, job control signals such as `SIGTSTP`, `SIGSTOP`, and `SIGCONT` cannot be used through these system calls.

> **Note:** The use of `tfork`(3), `t_fork`(3), `_lwp_kill`(2), and `_lwp_killm`(2) are not recommended because they may not be supported in future releases. It is suggested that you use the new Pthreads-based interfaces instead.

The specific changes to signal semantics are the following:

- The `kill`(2) and `killm`(2) system calls send a signal to a multitasking group, not the individual members of a multitasking group. The member that services the signal is chosen by the system.

  If an application uses `kill`(2) or `killm`(2), this change only affects the application if it is recompiled under the UNICOS 9.0 release. If an application is receiving a signal, the behavior is dependent on whether the sending application has been recompiled under UNICOS 9.0.

  This change adversely impacts only applications using the `tfork`(3) or `t_fork`(3) functions to create tasks.

- The `alarm`(2) system call sends a signal to any member of the multitasking group from which `alarm`(2) is called. Previously, the calling LWP received the signal. The old behavior is provided by using the `_lwp_alarm`(2) system call.

  > **Note:** Use of the `_lwp_alarm`(2) system call is not recommended because it may not be supported in future releases.

  This change affects only applications that are recompiled under UNICOS 9.0.

- The delivery of a signal that is asynchronous with respect to the event that caused the signal to be sent is delivered to any member of the multitasking group.

> **Caution:** Because the transmission of these asynchronous signals is not tightly coupled to the execution of a system call, this change impacts all multitasked applications, including existing binaries.

The change should adversely affect only applications using `tfork`(3) or `t_fork`(3) because Autotasking and macrotasking applications could never depend on any correspondence between tasks and LWPs.

The following behaviors are affected by this change:

- The delivery of the `SIGCLD` signal when a process exits.
- Signals used as parameters to the `reada`(2), `writea`(2), and `listio`(2) system calls.
- The delivery of the `SIGINFO` and `SIGCLULIM` signals upon such events as overrun of file system quotas and other limits.
- The delivery of the `SIGURG` and `SIGIO` signals for network I/O.
- The delivery of a signal used as a parameter to the `setjob`(2) system call.
- The delivery of a `SIGRECOVERY` signal after the restarting of a multitasking group.
- Signals generated by using terminal drivers (`SIGINT`, `SIGQUIT`, `SIGTSTP`, `SIGTTIN`, and `SIGTTOU`).
- Signals sent by using the `/proc PFCKILLM` interface.
- In `cdbx cont signo`, `signo` is a signal number. This `cdbx`(1) command now generates only one signal to a multitasking group. Previously, it generated one signal per LWP.

There are three primary reasons for changing this behavior for all these binaries:

- It would be difficult to track the binaries for which the old or new behavior would be appropriate.
- The new behavior is a more predictable and manageable behavior.
- The impact is expected to be very limited due to the limited scope of the `tfork`(3) and `t_fork`(3) functions.

3.11 Installation

The following subsections describe compatibility issues affecting the installation of the UNICOS operating system software.

3.11.1 *Installation and configuration tool interface changes*

Users affected

Administrator, operator, system analyst

Supporting hardware

All Cray Research systems

The following changes were made to the UNICOS Installation/Configuration Menu System interface:

- Incompatibility introduced with release: UNICOS 8.3

  Two rarely used keys have been eliminated in order to support the new X Window System interface; the `check key` command (`c` or `C`) and the ability of the accelerator key (`a` or `A`) to jump into form menus. This functionality is replaced in part by the ability to traverse the menu tree through a visual representation in the X Window System interface.

- Incompatibility introduced with release: UNICOS 9.0

  Configuration menu items that were disabled (marked N/A) are no longer visible as of the UNICOS 9.0 release; when the item is enabled, it will become visible.

3.11.2 *USM version restriction*

Users affected

Administrator, system analyst

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 8.3

Because UNICOS 9.0 requires a new UNICOS source manager (USM) feature to support named sideline branches, installers cannot use the UNICOS 8.0 version of USM to apply mods to a UNICOS 9.0 system.

Attempts to use the UNICOS 8.0 version of USM to apply a mod to a UNICOS 9.0 system will result in the following error message and the failure of the mod application:

```
$msg 374 The format for PL file file is invalid.
Please contact the database administrator.
```

3.11.3 *Media support for UNICOS 9.1 limited to CD–ROM*

Users affected

Administrators

Supporting hardware

All Cray Research systems except CRAY J90 and CRAY EL series

Incompatibility introduced with release: UNICOS 9.1

Beginning with UNICOS 9.1, CD–ROM will be the only supported release media. This will apply to all present and future platforms except CRAY J90 series and CRAY EL series.

3.12 Online diagnostics

The following incompatibility issues affect the online diagnostics software.

3.12.1 `olsbt` *limits CPU selection*

Users affected

Administrator, system analyst

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 8.3

To increase the efficiency of the `olsbt`(8) online diagnostic test, the number of CPUs that can be selected has been limited to 2 through 4 CPUs. The limitation has been imposed because the `olsbt`(8) test does not run efficiently on 1 CPU or on more than 4 CPUs. The `+anycpus` option allows a user to override the CPU restriction on a lightly loaded, dedicated system.

For more information, see the `olcfpt`(8) man page.

3.12.2 `olcfdt` *disk test support removed*

Users affected

Administrator, system analyst

Supporting hardware

All Cray Research systems

Incompatibility introduced with release: UNICOS 9.0

The `olcfdt` test was removed. All disk drives that were supported by the `olcfdt` test and any new disk devices are now supported by the `oldt`(8) disk test, which is an online confidence test for disks and mass storage.

For more information, see the `oldt`(8) man page.

3.12.3 *OLNET support removed for CRAY EL series*

Users affected

Administrator, system analyst

Supporting hardware

CRAY EL series

Incompatibility introduced with release: UNICOS 9.0

The OLNET online diagnostic network communications program is no longer supported for CRAY EL series.

3.12.4 *Final major release supporting* `olnet` *for HiPPI and FDDI*

Users affected

Administrator, system analyst

Supporting hardware

Cray PVP systems, except CRAY J90 series and CRAY EL series

Incompatibility introduced with release: UNICOS 10.0

Beginning with the UNICOS 10.0 release, `olnet`(8) support for HiPPI and FDDI will end. The equivalent diagnostic capabilities will be provided for HiPPI by `vht`(8) and for FDDI by `vft`(8) in the UNICOS 10.0 release.

Additional diagnostic support will be added for ATM by `vat`(8) and `vet`(8) for Ethernet. The remaining functionality of `olnet`(8), which includes FDR-4, NSC, FEI-3, and MPP I/O support will be in maintenance mode until hardware support for those devices ends. In maintenance mode, only critical and selected urgent bugfixes are provided.

Related Publications

- *`OLNET` Online Diagnostic Network Communications Program Maintenance Manual for UNICOS*, publication SD-1021

3.13 Disk and Memory Requirements

This section provides an approximation of disk and memory requirements for installing the UNICOS 9.0 release. Configuration parameters and machine characteristics influence binary sizes; therefore, this information provides only an estimation of what your site may experience.

3.13.1 *Disk requirements*

The following table displays disk requirements for UNICOS 9.0. The sizes represented in the table have been buffered by 15% and rounded up to the nearest thousand.

File system	Size in 4K blocks
`/`	82,000
`/usr`	174,000
`/tmp`	2,000
`/usr/src` (source)	376,000
`/usr/src` (binary)	286,000

These numbers are estimates, and partition size can vary greatly (especially `/src` and `/tmp`) depending on what the `NPROC` value is set to, what asynchronous products are loaded, and the number of user programs that will be put onto the system.

When these numbers were generated, a minimal `root` and `usr` file system was loaded, `NPROC` was set to four, and no asynchronous products were loaded. The maximum file size was found, 15 percent was added, and all file sizes were rounded up to the nearest thousand.

3.13.2 *Memory requirements*

This subsection describes the system size increase from the UNICOS 8.0 major release to the UNICOS 9.0 major release. The measurements were taken on a CRAY Y-MP2E system configured with 16MW of memory and identical configuration parameters and products for both UNICOS releases.

For the UNICOS 8.0 16 MW system, a sample configuration was created by first comparing the settings in the UNICOS installation/configuration menu system and then looking for the most common setting for the parameters from some customer sites that were surveyed. For the UNICOS 9.0 system, the configuration was kept consistent with that of the UNICOS 8.0 system.

The measurements for the amount of available user memory were gathered with the system running in multiuser mode. The `sysconf`(1) command was used to get the sizing numbers.

On the 16 MW system, the memory available to the user as compared to the total memory available decreased 4.1% from UNICOS 8.0 to UNICOS 9.0. The size decrease in user space will be approximately 650,000 words. This decrease will vary depending on site-configured products and parameters.

On larger systems, the amount of memory the system uses will increase, but the percentage of memory the system uses as compared to the total memory available will be smaller than what was measured on the 16 MW system. This is because the total memory used by the system will not increase proportionately to the total memory available.

Note: User programs that currently allocate all of the available user memory by using a hard-coded value may not be able to run because of the decrease in available user memory.